



**Karolinska
Institutet**

Using Git/GitHub in Scientific Collaborations

Stockholm 2021-01-26

Mikkel C. Vinding, PhD

NatMEG, CNS

Karolinska Institutet

Disclaimer: I have no affiliation with Git or GitHub, Inc.

Overview

- What is Git/GitHub?
- Starting a GitHub project
- Project workflow
- Collaborating
- Tutorial



```
$ git pull origin master
```

Code example

GitHub for (neuro)scientific collaboration

Why?

1. Organize your data analysis scripts
 2. Easy to collaborate on tasks that require programming (i.e. all neuroscientific data analysis)
 3. Share finished analysis scripts
-

Solution: use Git and GitHub

What is Git and GitHub?



- Tool for distributed version control
- Free and open source
- <https://git-scm.com/>



- Hosting service and interface for Git
 - Owned by Microsoft, but free*
 - Options for project management, collaboration, wikis , and more
 - <https://GitHub.com/>
-

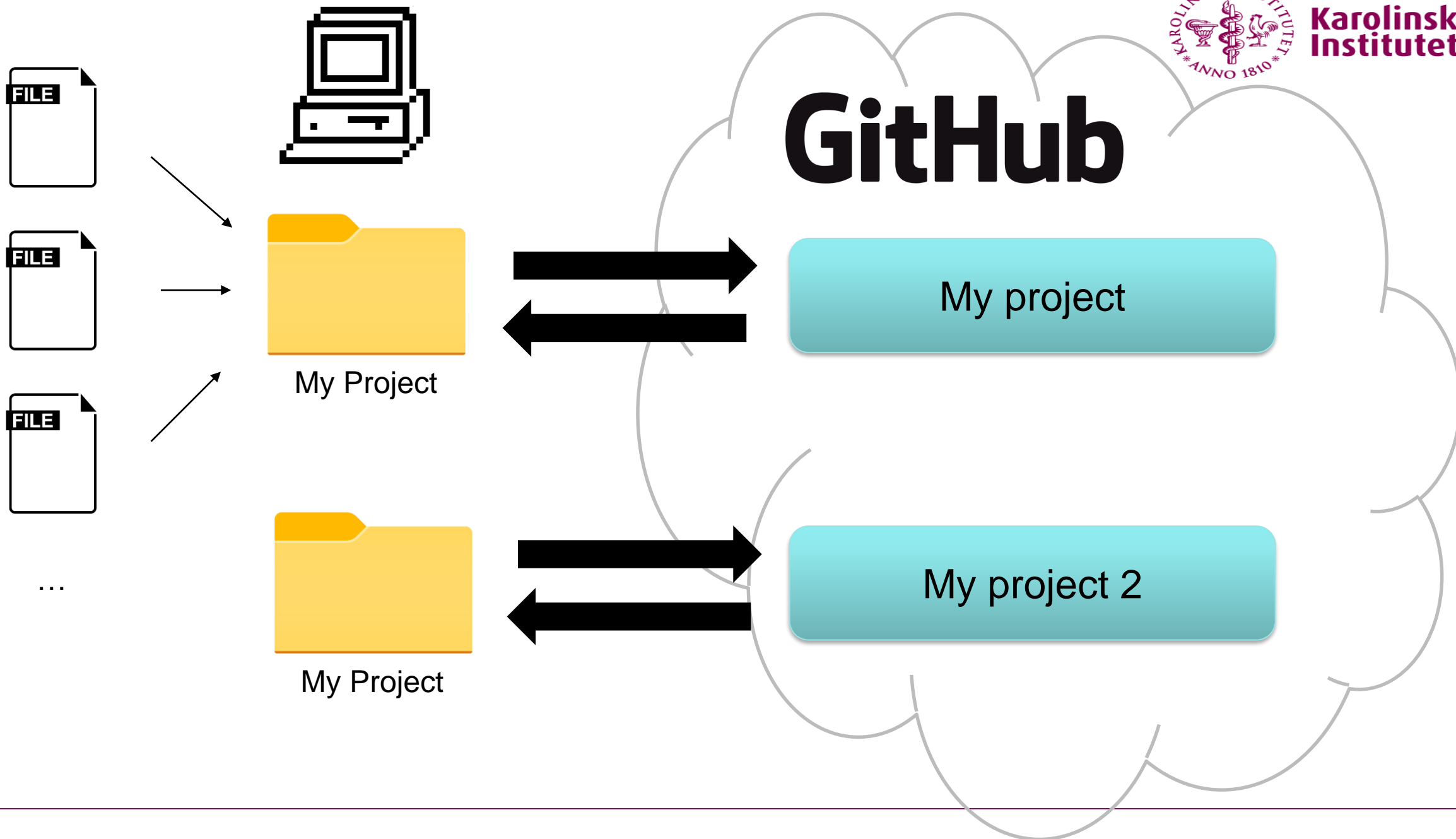
What NOT to put on GitHub

- Datafiles
 - Maximum file size is 100MB
 - Maximum repository size is 10GB
- Sensitive information



Subjid	Name	Identity_nr	Address
0001	Mikkel Vinding	123456-7890	Nobels väg 9
0002	...		

GIT WORKFLOW

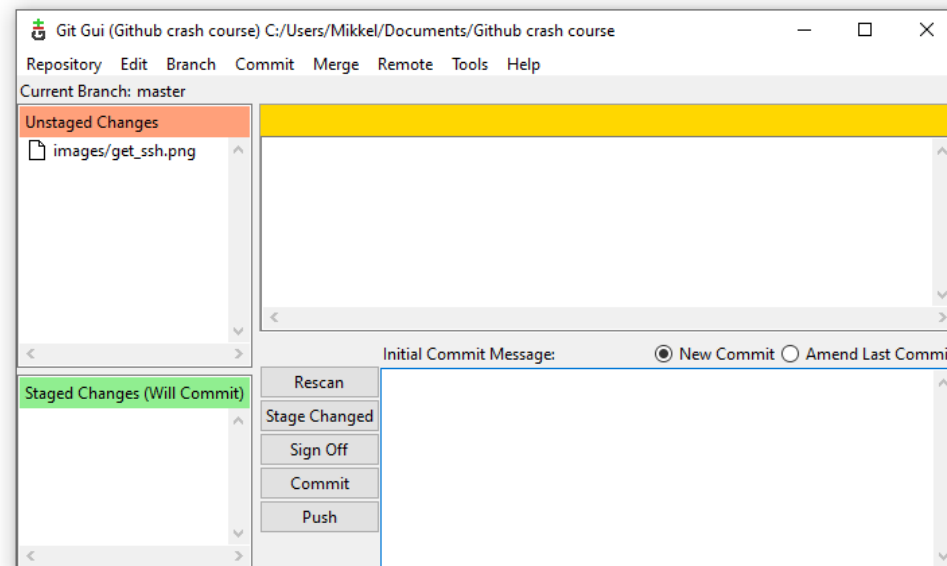


Using Git

```
$ git <command> <option(s)> <files> <...>
```

Do I have to use the terminal?

Yes*



*No: There is a GUI

START A GIT PROJECT

Getting started...

Start a new project

Create a repository for your project.

Where you want to start!

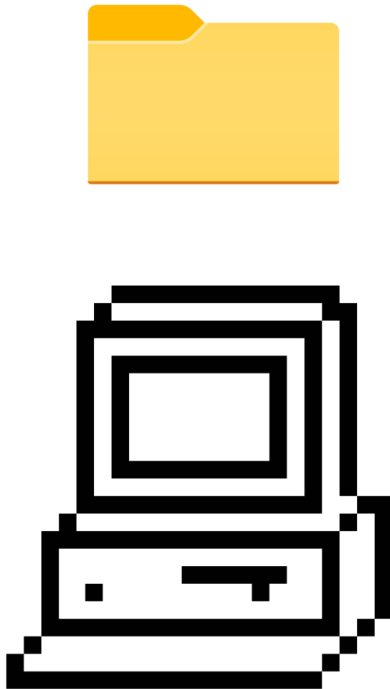
Join a project

E.g. a project where someone already have created a repository

Clone a project

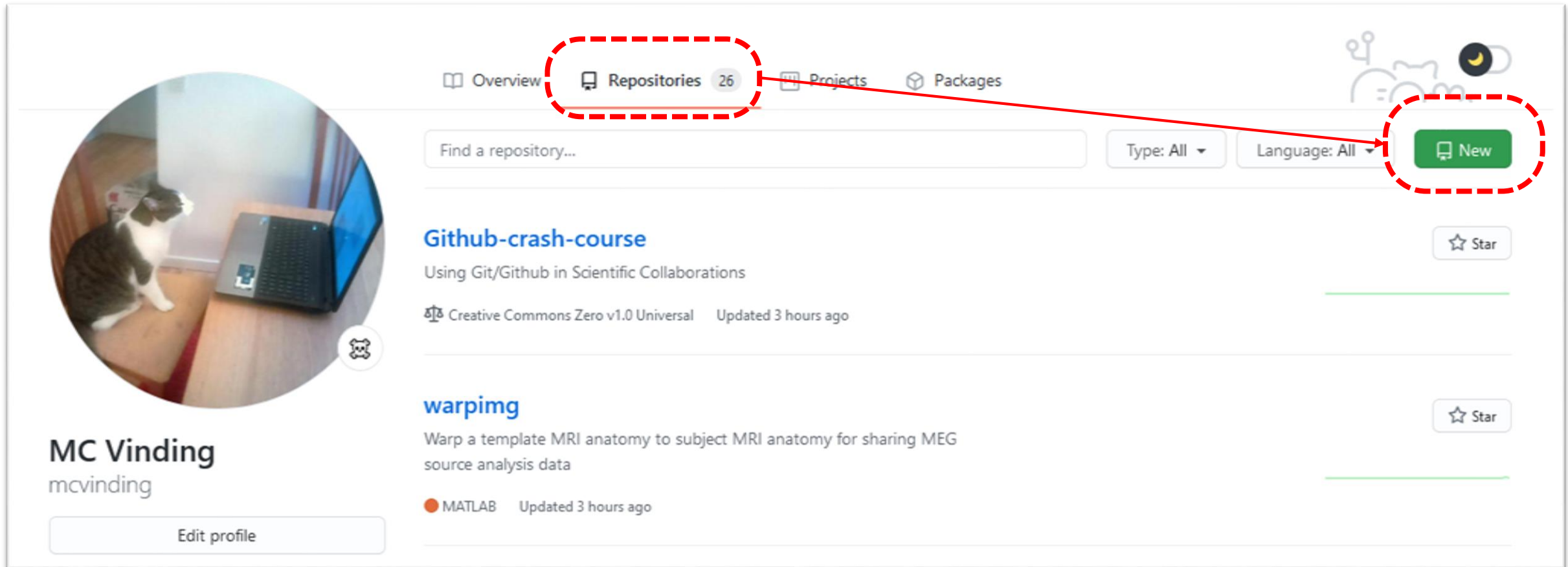
Get code from another project, e.g. an analysis toolbox

Start a new project...



GitHub

Start a new project...



Overview **Repositories 26** Projects Packages

Find a repository... Type: All Language: All **New**

Github-crash-course
Using Git/Github in Scientific Collaborations
Creative Commons Zero v1.0 Universal Updated 3 hours ago Star

warping
Warp a template MRI anatomy to subject MRI anatomy for sharing MEG source analysis data
MATLAB Updated 3 hours ago Star

MC Vinding
mcvinding
Edit profile

Create a new repository




Karolinska
Institutet

1. Name

Owner *



Repository name *

 mcvinding /

Great repository names are short and memorable. Need inspiration? How about [glowing-octo-pancake?](#)

Description (optional)

2. Define visibility

- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

3. ?????

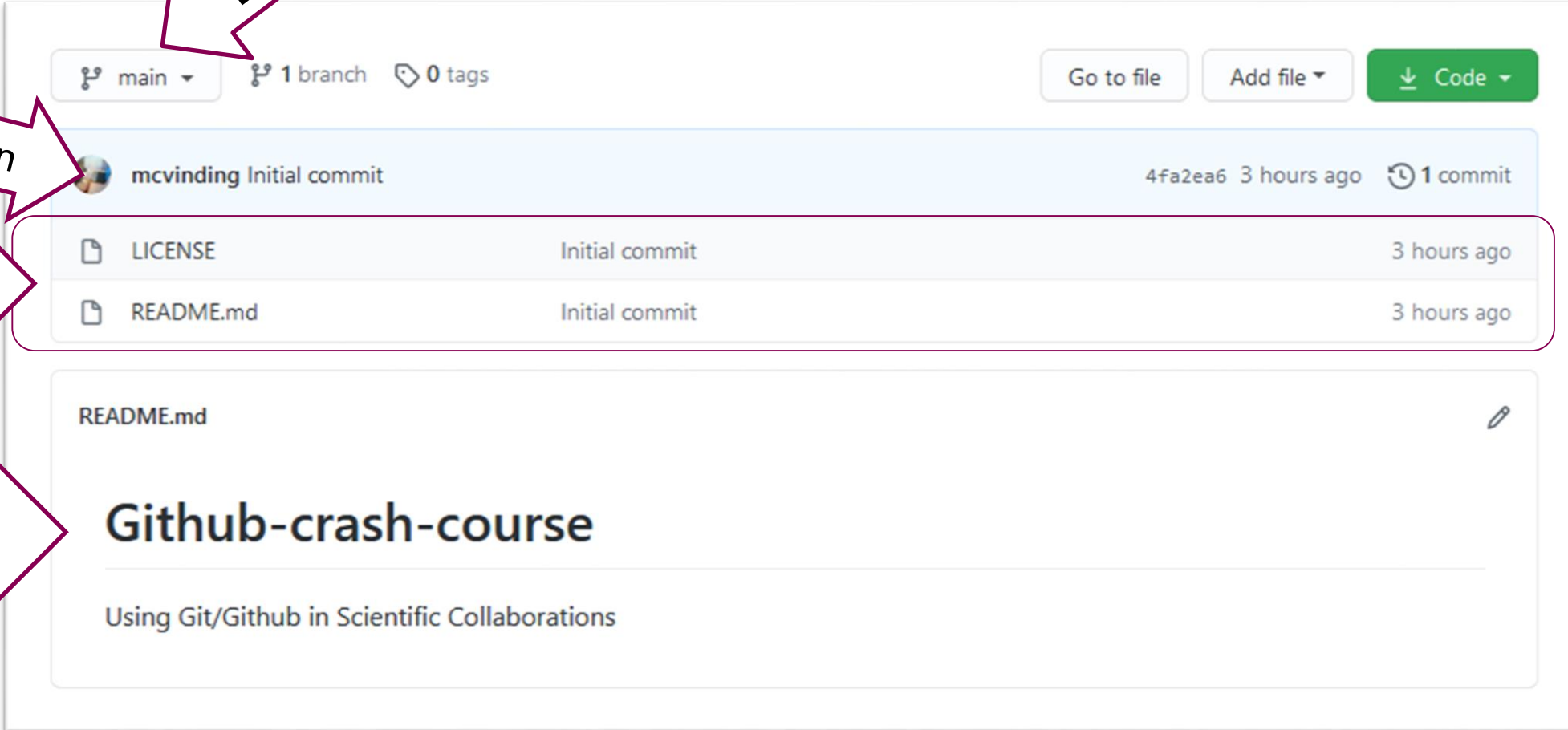
Initialize this repository with:

Skip this step if you're importing an existing repository.

- ☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)
- ☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)
- ☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

4. Create

Create repository



The screenshot shows a GitHub repository interface. At the top, there's a branch selector showing 'main' with a dropdown arrow, followed by '1 branch' and '0 tags'. To the right are buttons for 'Go to file', 'Add file', and a green 'Code' button. Below this is a commit header for 'mcvinding Initial commit' with commit hash '4fa2ea6', time '3 hours ago', and '1 commit'. A table lists files: 'LICENSE' and 'README.md', both marked as 'Initial commit' and '3 hours ago'. Below the file list, the 'README.md' content is displayed, featuring the title 'Github-crash-course' and the subtitle 'Using Git/Github in Scientific Collaborations'. Four purple arrows with text labels point to specific parts of the interface: 'Branch' points to the branch selector, 'Current version' points to the commit header, 'Your files' points to the file list table, and 'README.md' points to the README content area.

Branch

Current version

Your files

README.md

main 1 branch 0 tags

Go to file Add file Code

mcvinding Initial commit 4fa2ea6 3 hours ago 1 commit

LICENSE	Initial commit	3 hours ago
README.md	Initial commit	3 hours ago

README.md

Github-crash-course

Using Git/Github in Scientific Collaborations

The README file

- Write *informative text* that help people who find your repository
 - Collaborators...
 - Reviewers...
 - Public...
- Edit in text editor or in browser
- Markdown syntax

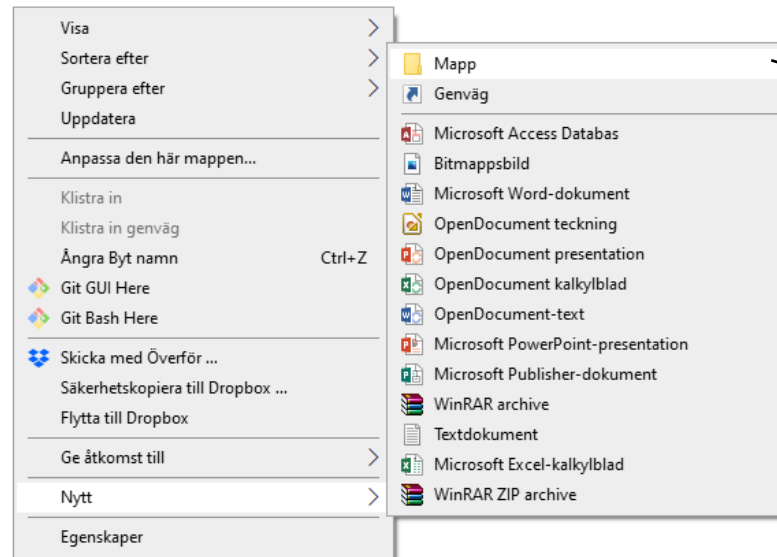


Nice Markdown cheatsheet:
<https://GitHub.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

Start a new project...

- Create a folder at the desired location

```
$ mkdir my_project
```



My Project

Start a new project...

- Create a folder at the desired location
- Go to folder

```
$ cd my_project
```



My Project

Start a new project...

- Create a folder at the desired location
- Go to folder
- Initialize folder

```
$ git init  
Initialized empty Git repository in  
C:/Users/Mikkel/Documents/GitHub crash course/.git/
```



My Project

Start a new project...

- Create a folder at the desired location
- Go to folder
- Initialize folder
- Set remote

GitHub



My Project

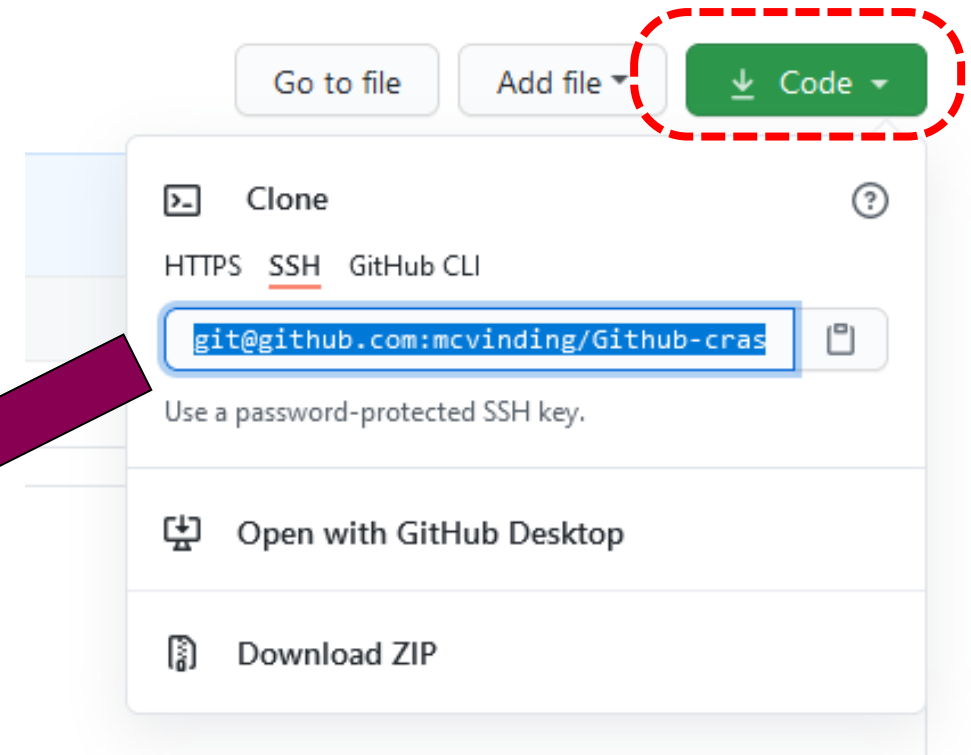


My Project

Start a new project...

- Create a folder at the desired location
- Go to folder
- Initialize folder
- Set remote

```
$ git remote add origin <address>
```

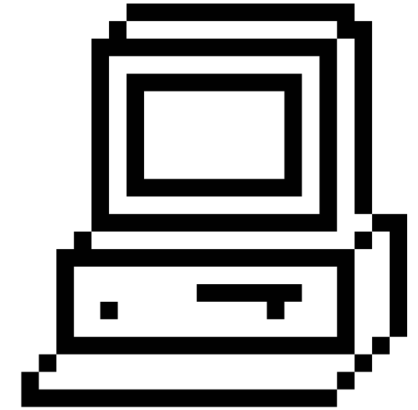


Start a new project...

- Create a folder at the desired location
- Go to folder
- Initialize folder
- Set remote

```
$ git remote add origin <address>
```

GitHub



My Project
"origin"



My Project



See remote address

```
$ git remote -v  
origin  git@GitHub.com:mcvinding/GitHub-crash-course.git (fetch)  
origin  git@GitHub.com:mcvinding/GitHub-crash-course.git (push)
```


Start a new project...

- Create a folder at the desired location
- Go to folder
- Initialize folder
- Set remote
- Get files from remote: "pull"

```
$ git pull origin main
```

"Branch"

GitHub



My Project
"origin"



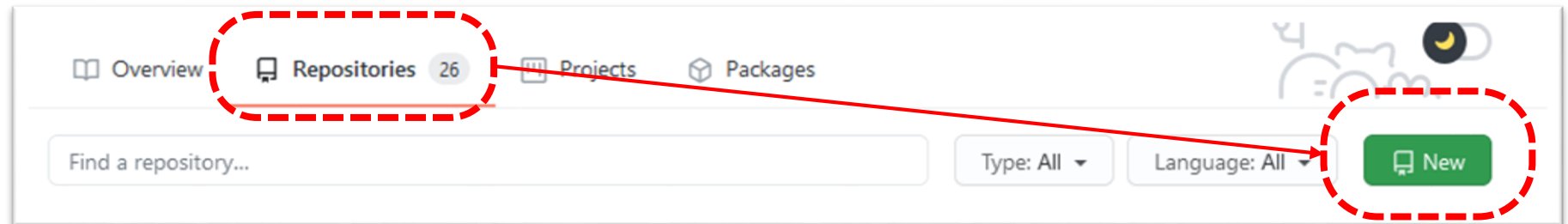
My Project

Pull files from remote

```
$ git pull origin main
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
From GitHub.com:mcvinding/GitHub-crash-course
* branch          main          -> FETCH_HEAD
```

Summary: initiate a Git project

GitHub



```
$ git init  
$ git remote add origin <address>  
$ git pull origin main
```

Getting started...

Start a new project

Create a repository for your project.

Where you want to start!

Join a project

E.g. a project where someone already have created a repository

Clone a project

Get code from another project, e.g. an analysis toolbox

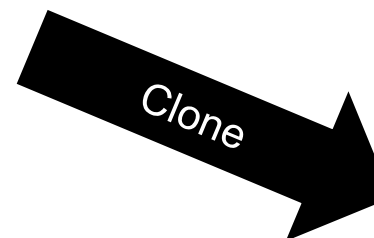
Git clone

```
$ git clone <address>
```

GitHub



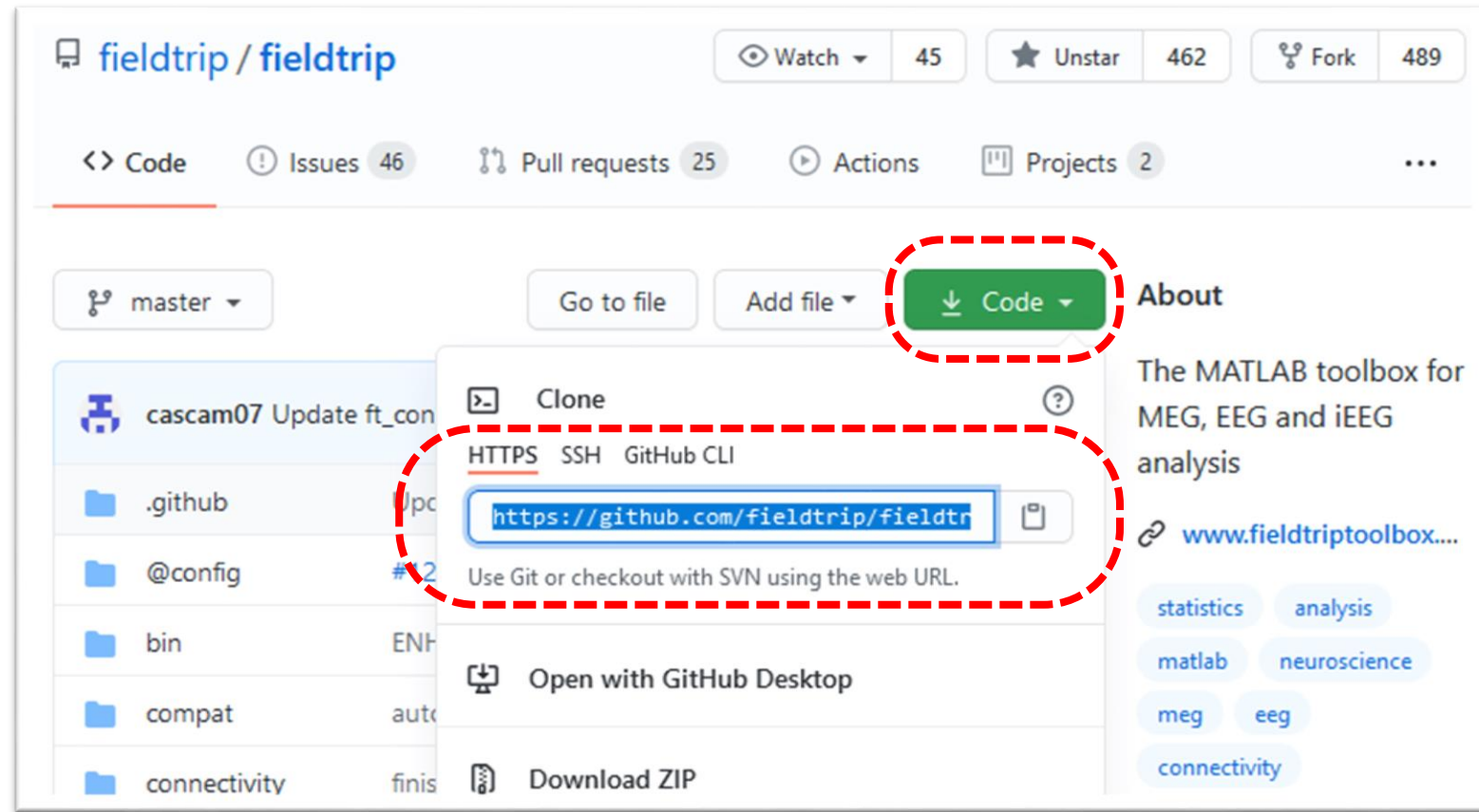
Any GitHub repo



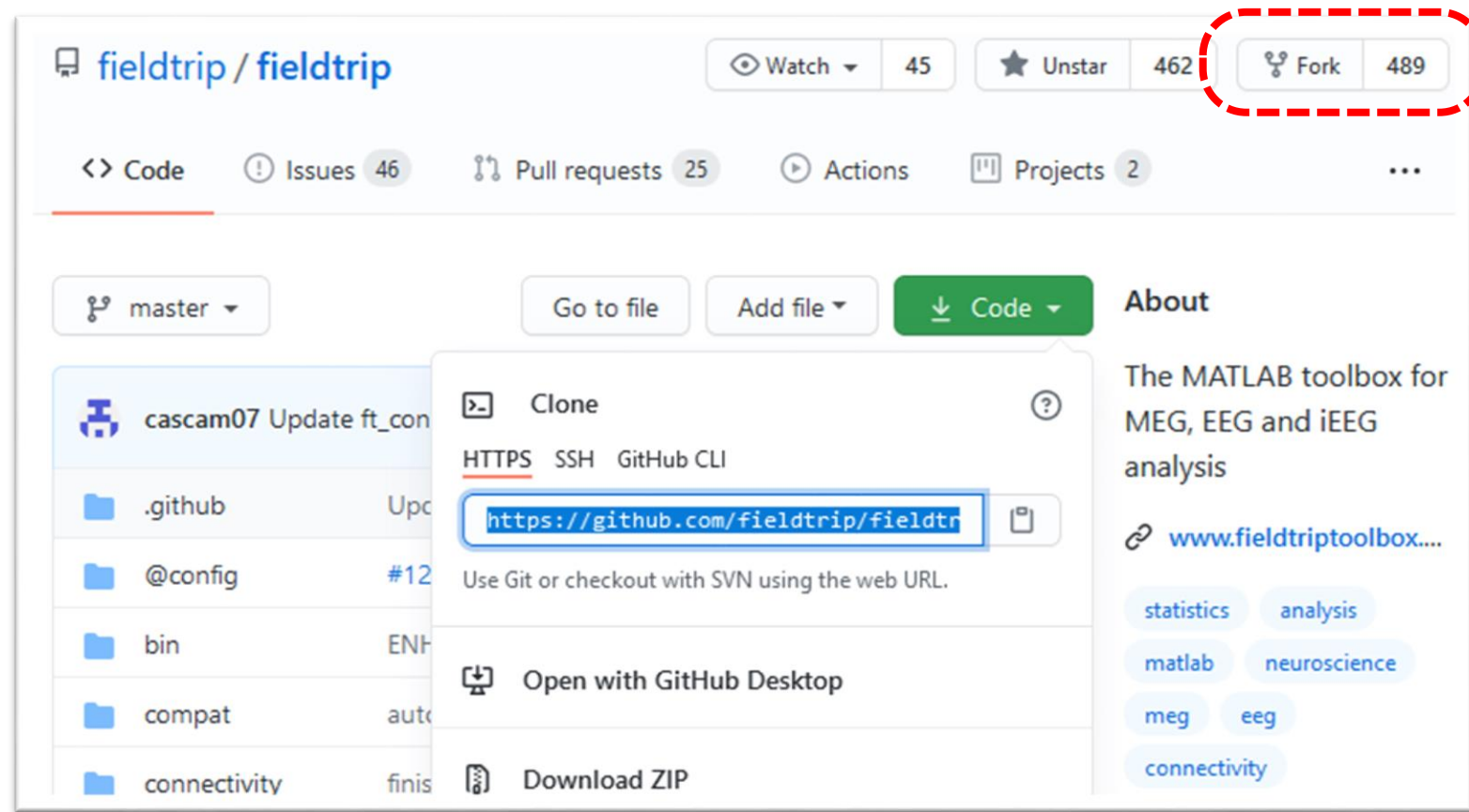
My local copy

Git clone

```
$ git clone <address>
```



Fork

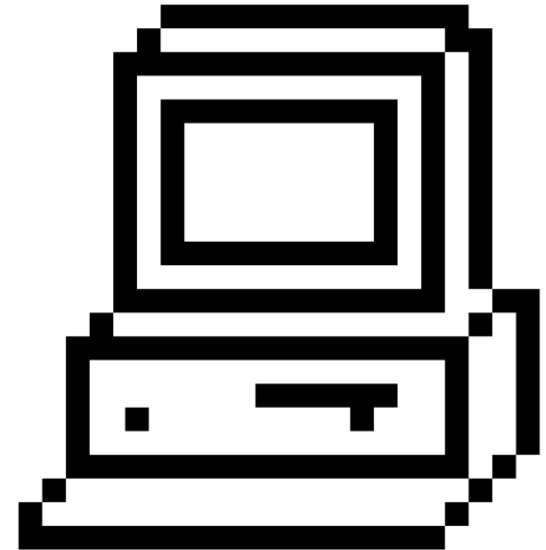
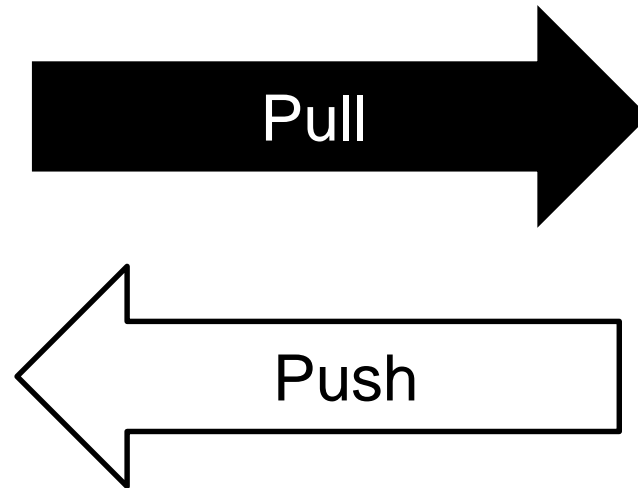


Pull, commit, push, and win at code management

WORKING WITH GIT REPOSITORIES

Git terminology

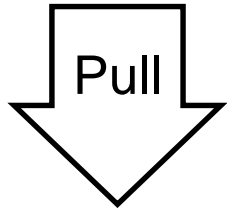
GitHub



Git workflow at a glance

- **Pull** latest code
 - Write your code...
 - Stage edits
 - Commit changes
 - **Push**
-

GitHub



```
$ git pull <remote> <branch>
```

GitHub

Pull



Edit your
files...

Local stage

```
$ git add <filename1> <filename2> <...>
```



myFile1.m



myFile2.py



myFile3.txt

The most used Git command

```
$ git status
```

➡ \$ git status
On branch main
Changes to be committed:
(use "git reset HEAD <file>..." to unstage)

➡ new file: new_file.txt

Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git checkout -- <file>..." to discard changes in working directory)

➡ modified: another_new_file.txt

➡ Untracked files:
(use "git add <file>..." to include in what will be committed)

➡ images/

Local stage



"Off stage"

"Not even..."

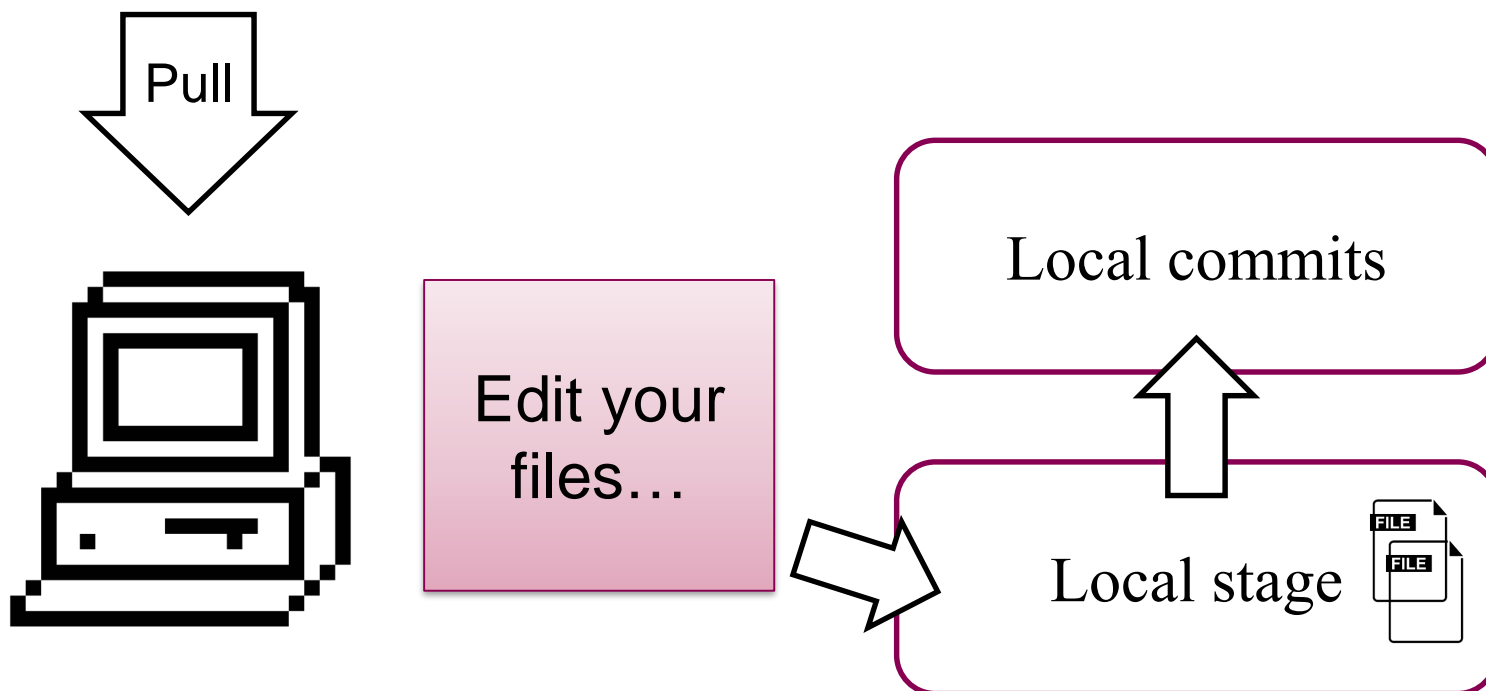
Staging files

Specific files

```
$ git add <filename1> <filename2> <...>  
$ git add -u
```

All tracked but unstaged files

GitHub

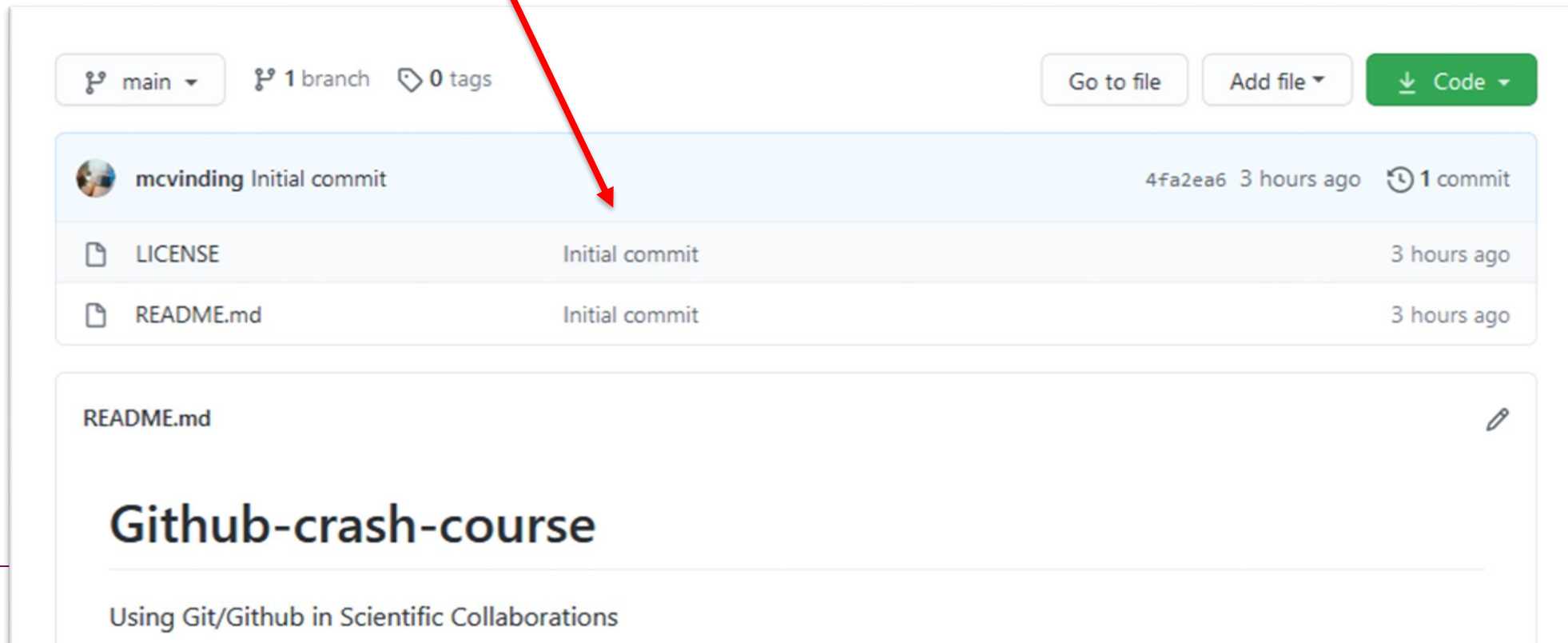


```
$ git commit -m "<...>"
```


Commit messages

```
$ git commit -m "<your message>"
```

Write a short informative
text about changes



The screenshot shows a GitHub repository page for 'Github-crash-course'. At the top, there are buttons for 'Go to file', 'Add file', and 'Code'. Below this, a commit by 'mcvinding' is shown with the message 'Initial commit'. The commit hash is '4fa2ea6' and it was made '3 hours ago'. Below the commit, a list of files is shown: 'LICENSE' and 'README.md', both with the message 'Initial commit' and '3 hours ago'. At the bottom, the 'README.md' content is visible, starting with 'Github-crash-course' and 'Using Git/Github in Scientific Collaborations'.

File	Commit Message	Time
LICENSE	Initial commit	3 hours ago
README.md	Initial commit	3 hours ago

Commit messages

```
$ git commit -m "<your message>"
```

Write a short informative
text about changes

Too little

"Some changes"

"Committed stuff"

OK

"changed filter settings"

"added files for plots"

"overhaul of preprocessing"

Too much

*"changed filter settings in
line 42-46 and added a
bunch more options for
processing and new scripts
for plots with pretty colours
that look like a pretty
flower"*

GitHub

Pull

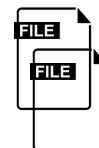


Edit your
files...

Push

Local commits

Local stage



```
$ git push <remote> <branch>
```

GitHub

Pull

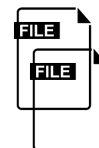


Edit your
files...

Push

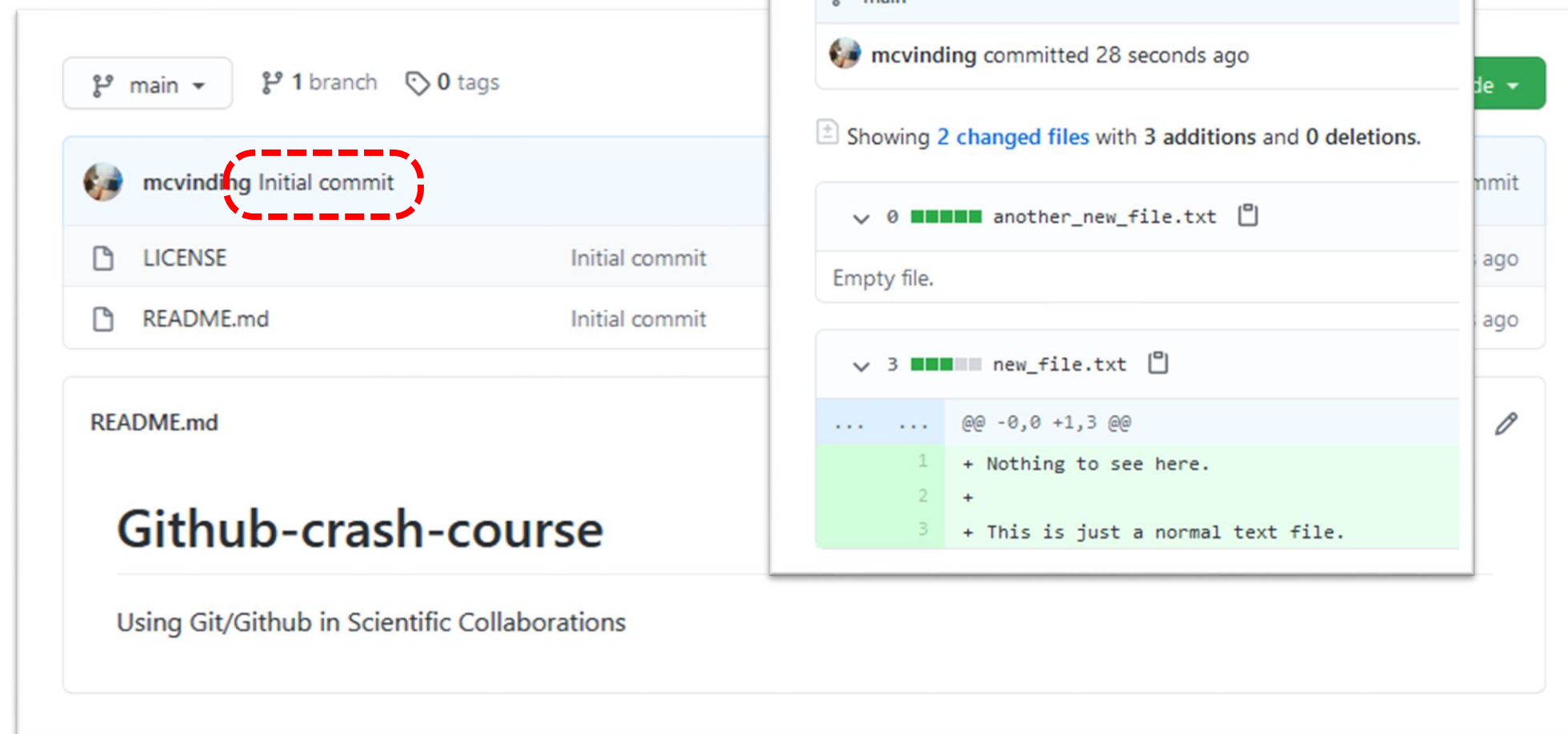
Local commits

Local stage



```
$ git push origin main
```

More on Git commits



main 1 branch 0 tags

mcvinding Initial commit

LICENSE Initial commit

README.md Initial commit

README.md

Github-crash-course

Using Git/Github in Scientific Collaborations

added new_file

main

mcvinding committed 28 seconds ago

Showing 2 changed files with 3 additions and 0 deletions.

0 another_new_file.txt

Empty file.

3 new_file.txt

```
... @@ -0,0 +1,3 @@
1 + Nothing to see here.
2 +
3 + This is just a normal text file.
```

Git workflow

- **Pull** latest code
- Write your code...
- Stage edits (**add**)
- **Commit** changes
- **Push**

```
$ git pull origin main
```

```
$ git add <options>
```

```
$ git commit -m "..."
```

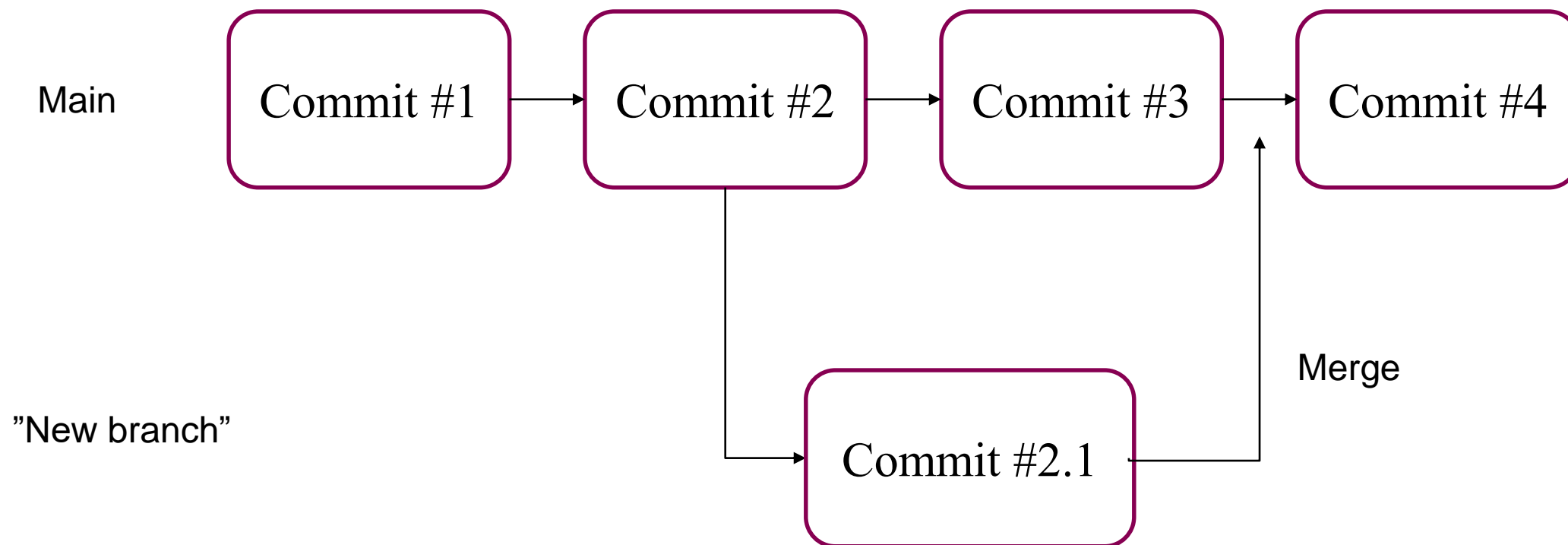
```
$ git push <remote> <branch>
```

How often should I commit?

How often you want

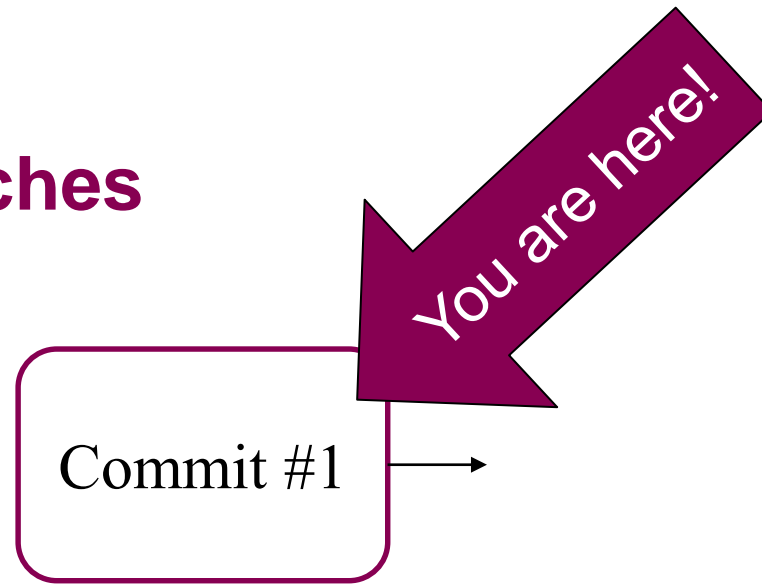
GIT BRANCHES

Branches



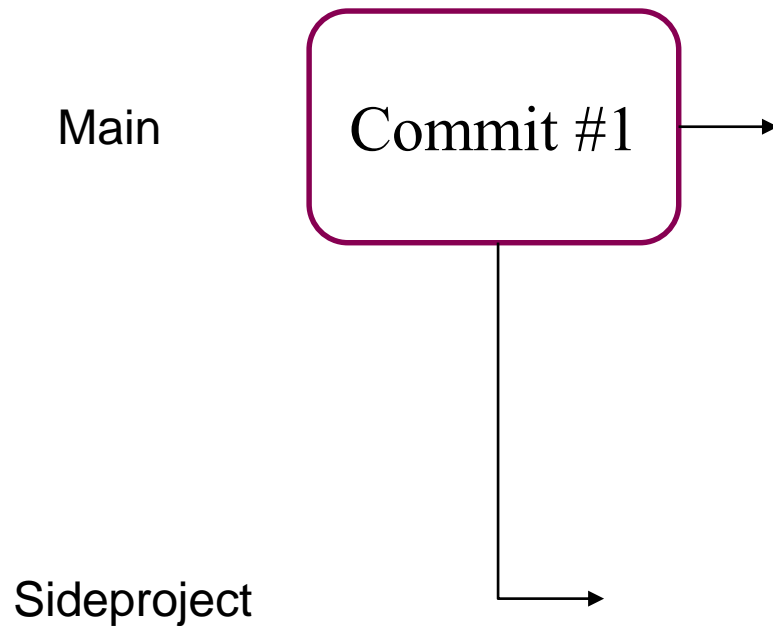
Branches

Main



```
$ git branch  
* Main
```

Branches

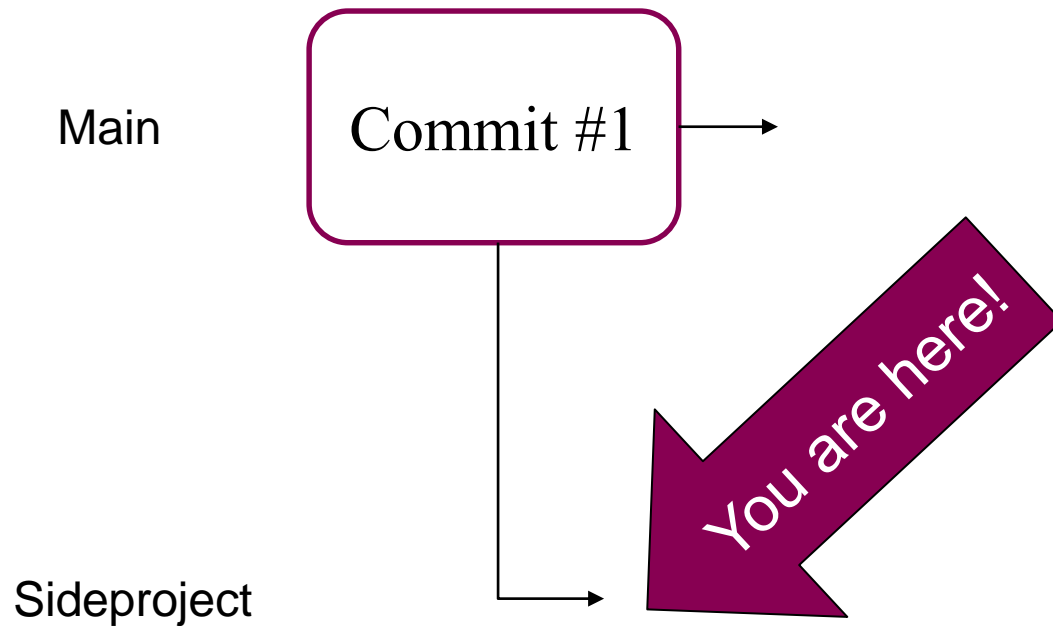


```
$ git branch
* Main
$ git checkout -b sideproject
```

Create new flag

Name of new branch

Branches

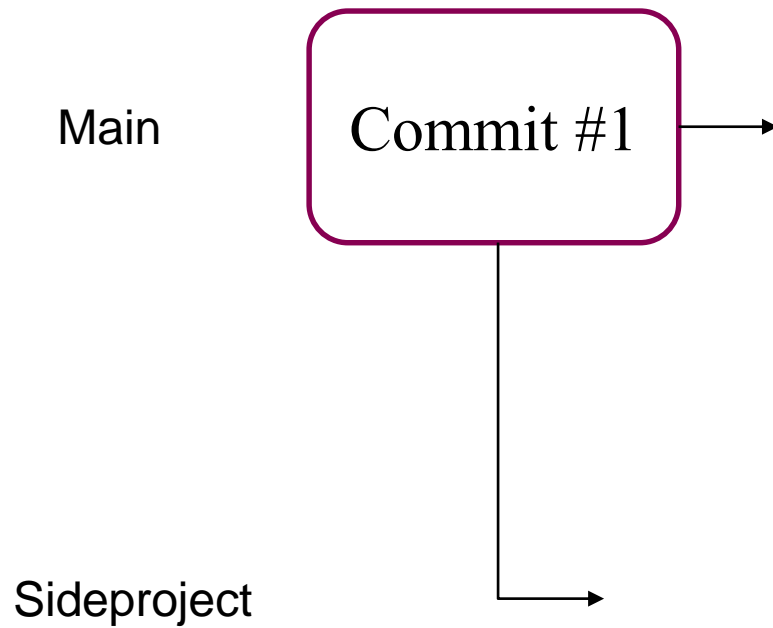


```
$ git branch
* Main

$ git checkout -b sideproject

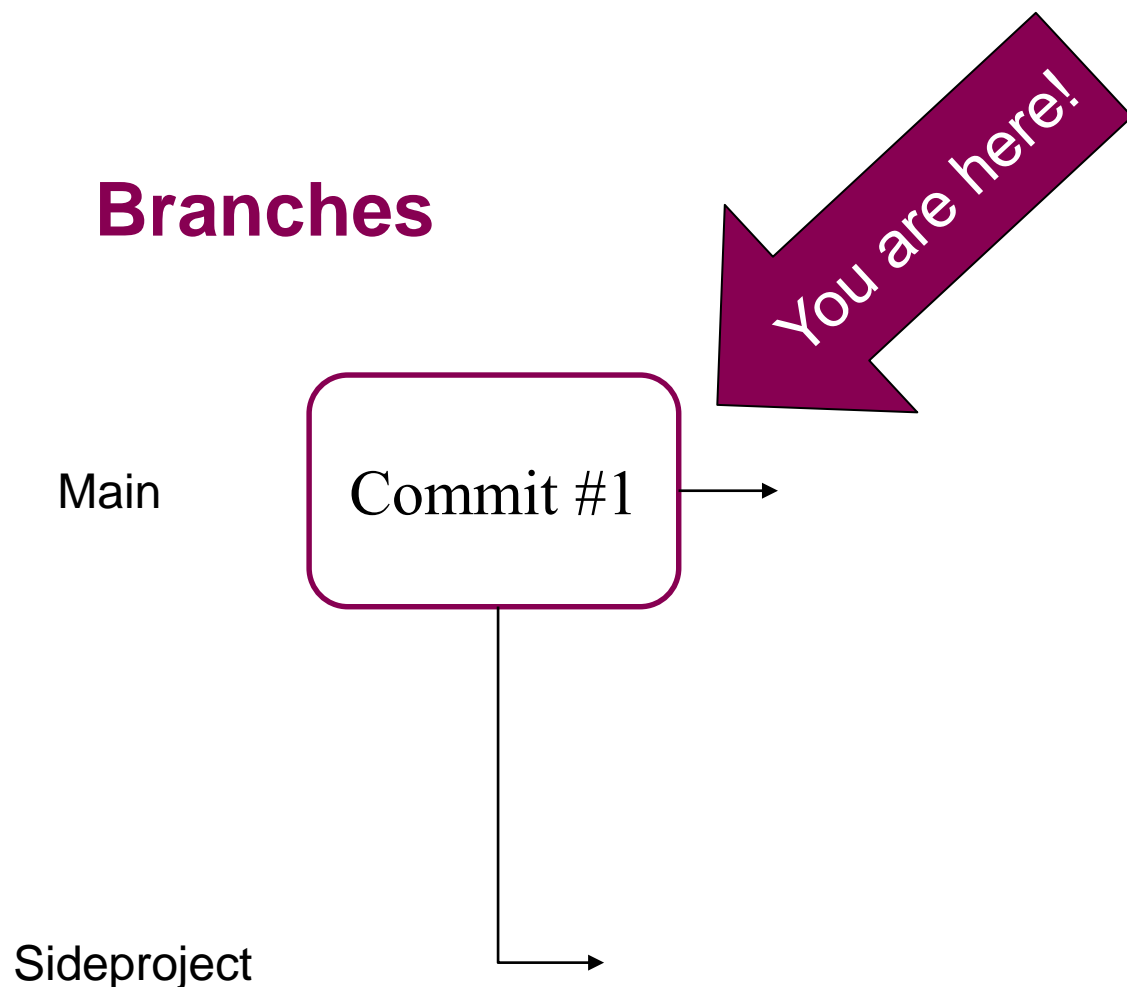
$ git branch
main
* sideproject
```

Branches



```
$ git commit -m "<...>"  
$ git push origin sideproject
```

Branches

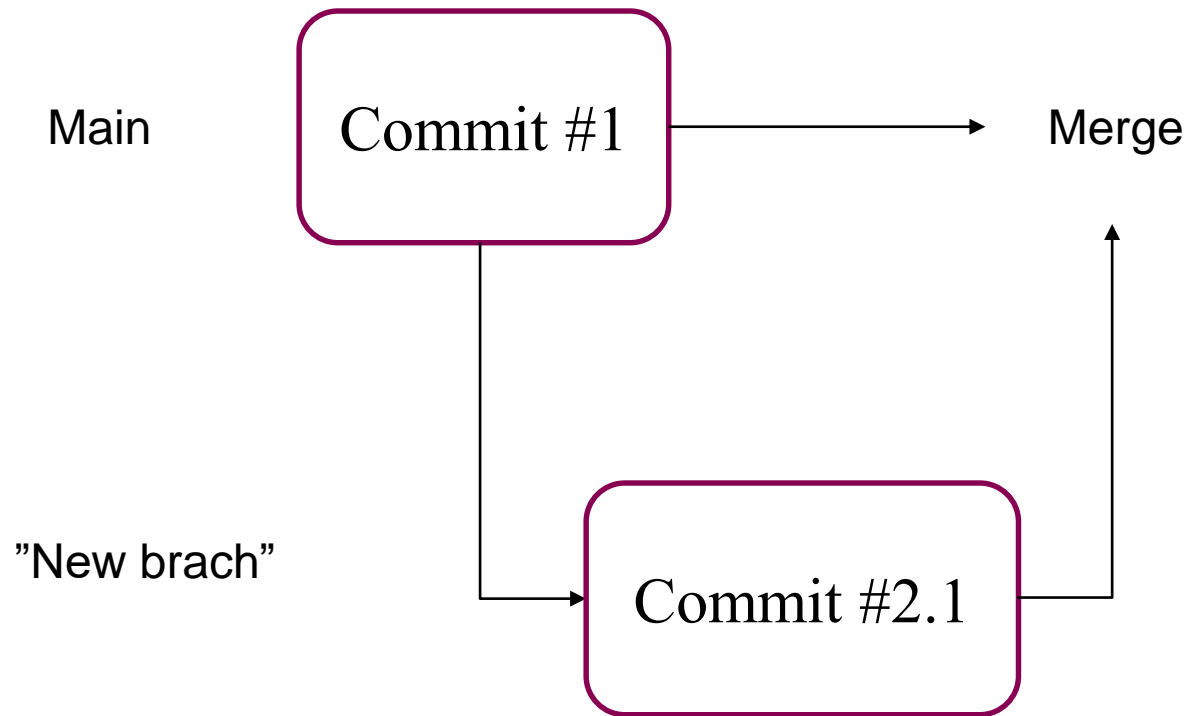


No "-b" flag



```
$ git checkout main  
$ git branch  
* main  
  sideproject
```

Merge







```
$ git checkout main  
$ git merge sideproject
```


Alternative merge


```
$ git push origin sideproject
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'sideproject' on GitHub by visiting:
remote:      https://GitHub.com/mcvinding/GitHub-crash-
course/pull/new/sideproject
remote:
To GitHub.com:mcvinding/GitHub-crash-course.git
 * [new branch]      sideproject -> sideproject
```


Alternative merge

 sideproject had recent pushes less than a minute ago [Compare & pull request](#)

 main  3 branches  0 tags [Go to file](#) [Add file](#) [Code](#)

 mcvinding added new_file 9fb4a7c 33 minutes ago ⌚ 2 commits

 LICENSE Initial commit 6 hours ago

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



base: main ▾



compare: sideproject ▾

✓ **Able to merge.** These branches can be automatically merged.



two new files

Write

Preview


H B I ≡ <> 🔗 ≡ ≡ ☑ @ ↗ ↶

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.



Create pull request ▾

 Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

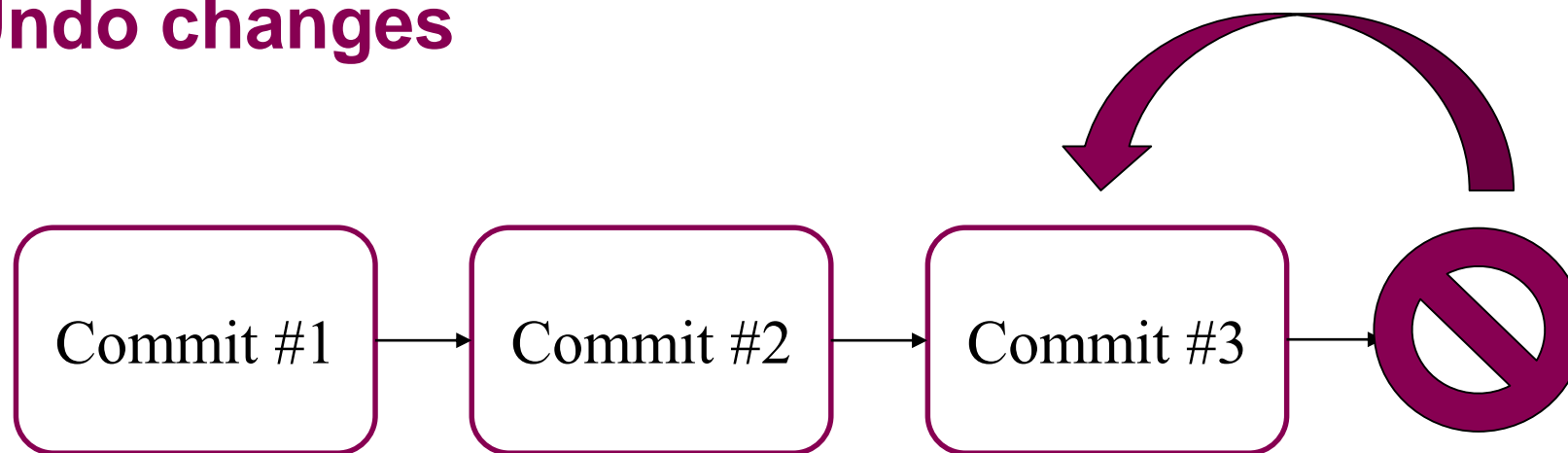
When to use branches

- When implementing entirely new features/analysis parts
- Testing out new code (and don't want to risk ruining the old)
- More useful when collaborating and delegating tasks

When not to use branches (stay on main)

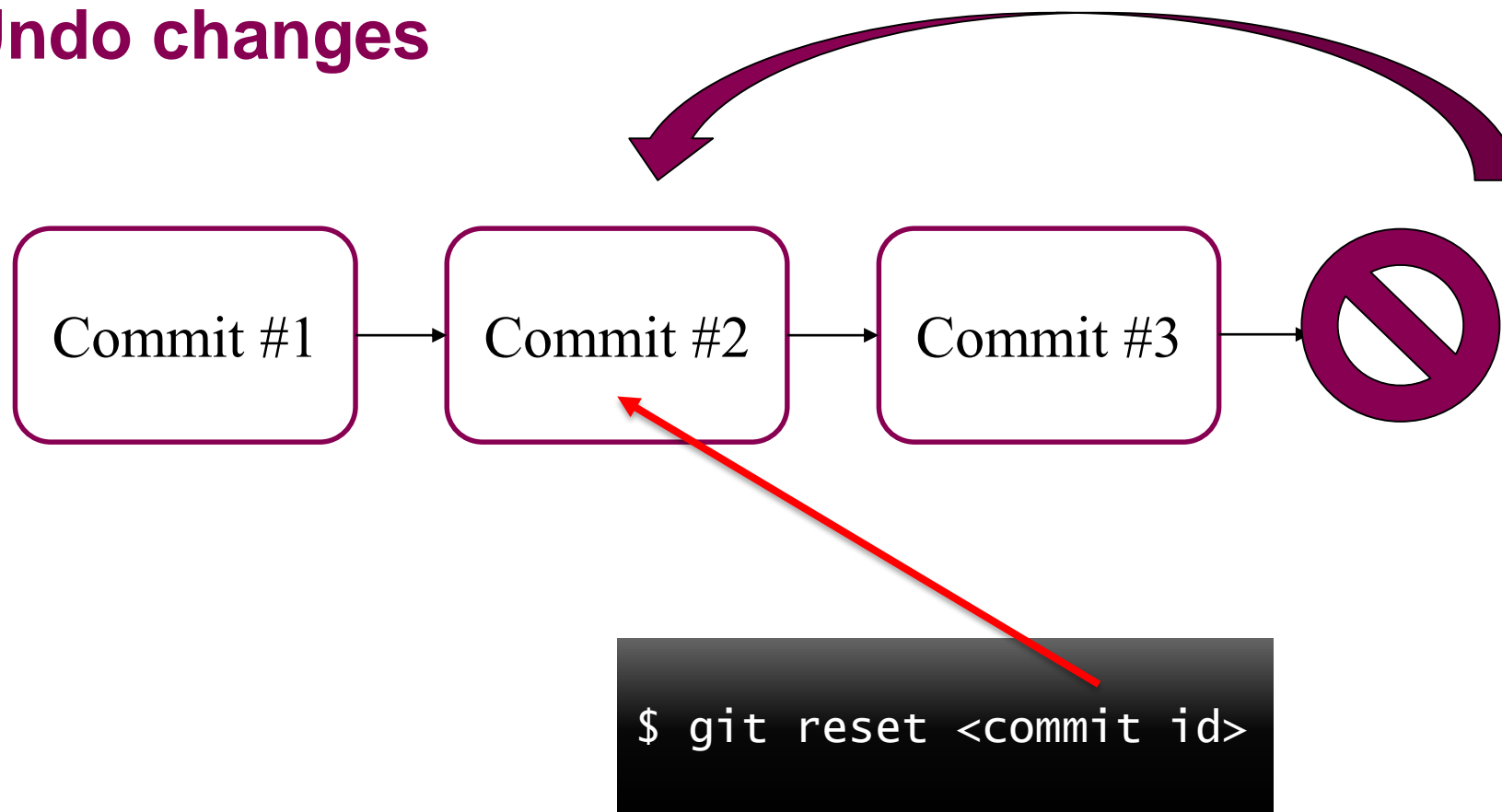
- Changes you will do anyway
 - To scrap the current project
-

Undo changes

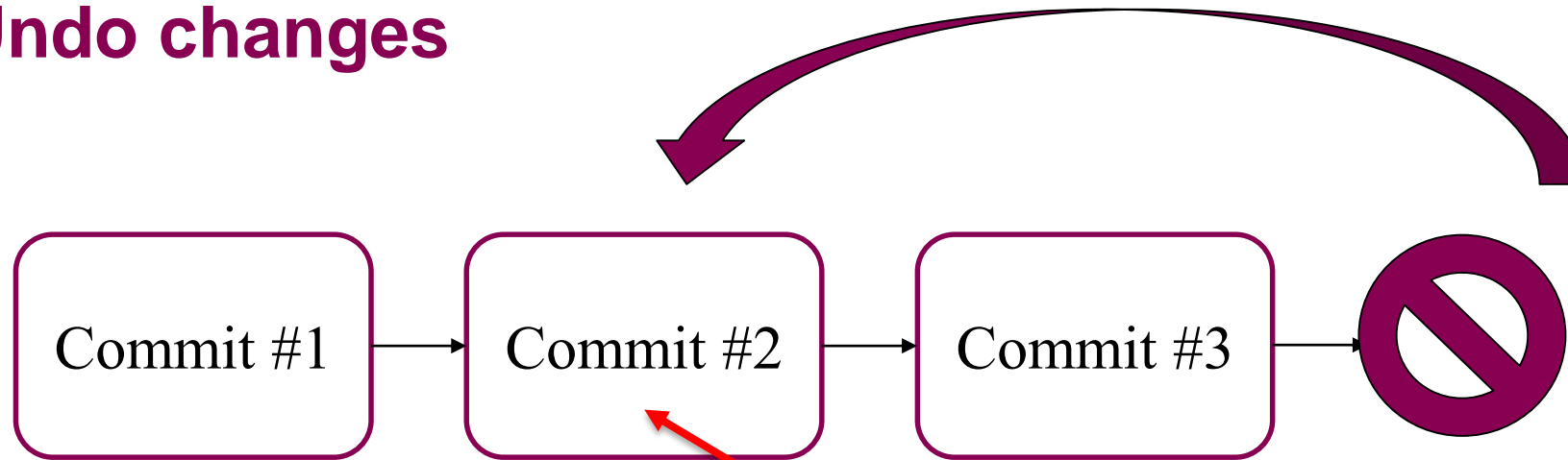


```
$ git reset --hard HEAD
```

Undo changes



Undo changes



```
$ git reset 4fa2ea669e898f2ba0549f50eada713ccd65cf89
```

```
$ git log
commit 9fb4a7cbe74a87abccb5fd3fb1d04e529f3417bb (HEAD ->
main, origin/main)
Author: mcvinding <mikke1.vinding@gmail.com>
Date:   Mon Jan 20 16:33:56 2021 +0100

    added new_file

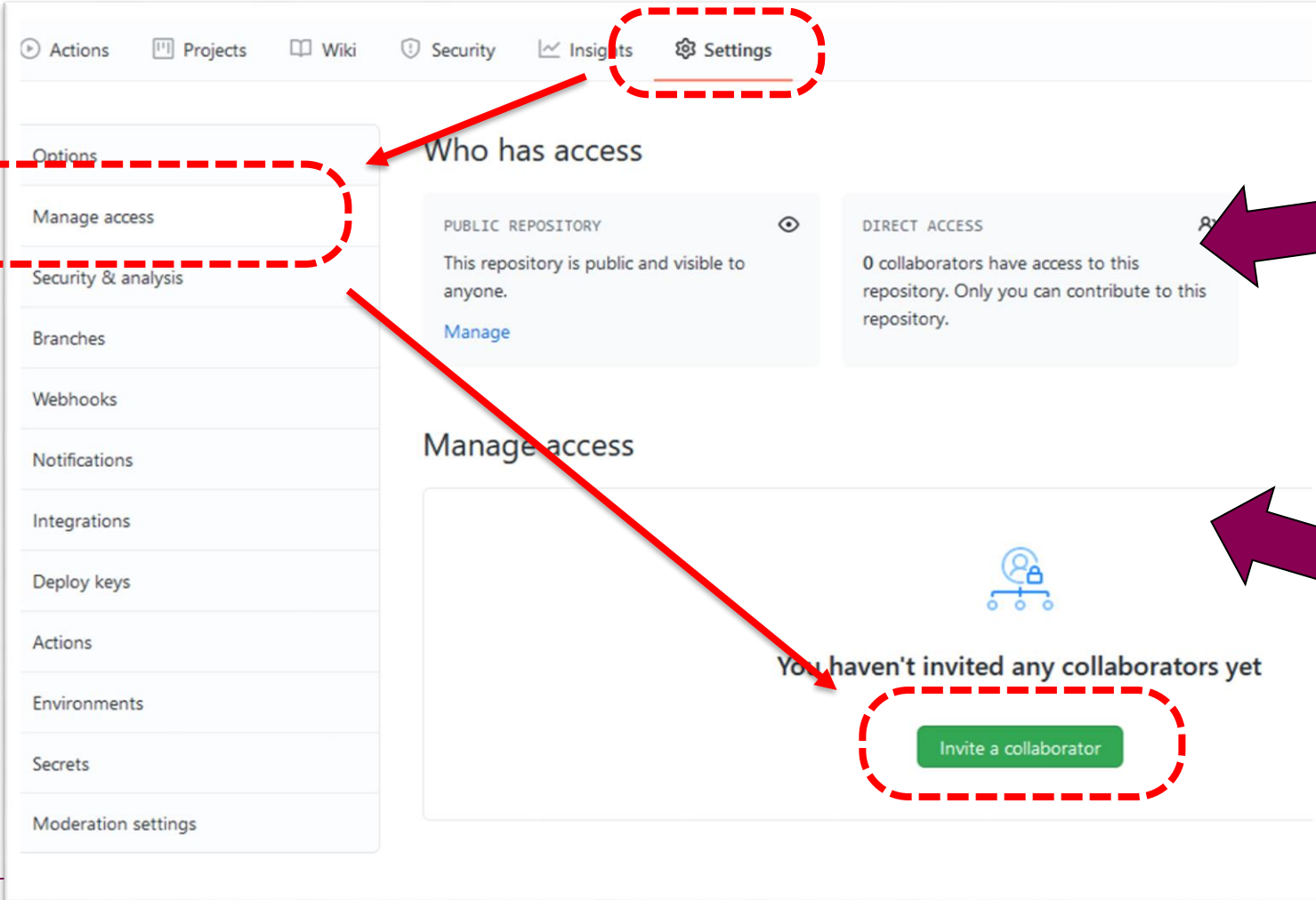
commit 4fa2ea669e898f2ba0549f50eada713ccd65cf89
(origin/master, main)
Author: MC Vinding <mikke1.vinding@gmail.com>
Date:   Mon Jan 20 16:26:53 2021 +0100

    initial commit
```

The more, the merrier: using GitHub with your lab mates

GITHUB COLLABORATIONS

Adding collaborators



The screenshot shows the GitHub repository settings page. The 'Settings' tab is selected and highlighted with a red dashed box. In the left sidebar, the 'Manage access' option is highlighted with a red dashed box. A red arrow points from this option to the 'Who has access' section. The 'Who has access' section shows 'PUBLIC REPOSITORY' and 'DIRECT ACCESS' with '0 collaborators'. A purple arrow points from this section to the 'Manage access' section. The 'Manage access' section shows a message 'You haven't invited any collaborators yet' and a green 'Invite a collaborator' button, which is highlighted with a red dashed box. A red arrow points from the 'Manage access' section to this button. A purple arrow points from the 'Manage access' section to the 'Invite a collaborator' button.

Actions Projects Wiki Security Insights **Settings**

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Actions

Environments

Secrets

Moderation settings

Who has access

PUBLIC REPOSITORY

This repository is public and visible to anyone.

[Manage](#)

DIRECT ACCESS

0 collaborators have access to this repository. Only you can contribute to this repository.

Manage access

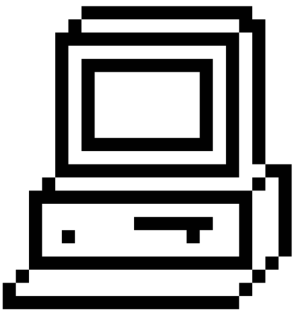
You haven't invited any collaborators yet

[Invite a collaborator](#)

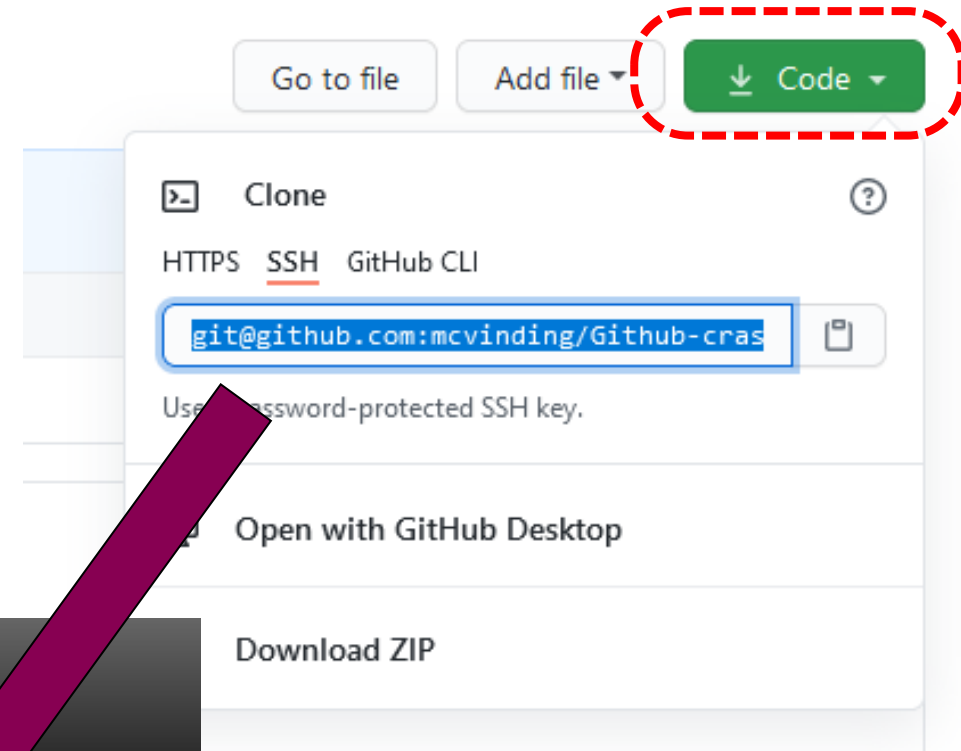


Setting up collaboration

- Initialize folder on your computer
- Go to your collaborators project
- Add remote collaborators remote



```
$ git init  
$ git remote add origin <address>  
$ git pull origin main
```



Collaborating (same procedure as before)

- **Pull** latest code
- Write your code...
- Stage edits
- Commit changes
- **Push**

```
$ git pull origin main  
$ git add <options>  
$ git commit  
$ git push <remote> <branch>
```

Summary

- Use Git/GitHub to manage your analysis scripts (not data).
- Basic usage (same if you are working on your own or in collaboration)

```
$ git remote add <remote> <branch>  
$ git pull origin main  
$ git add <options>  
$ git commit  
$ git push <remote> <branch>
```